Lightweight Rice Planthopper Identification Method Based on YOLOv5

Siquan Li, Yi Wang, Teng Shi, Xi Chen, Zhen Tang, Ziyu Zeng, Xin Wen, Yanling Shang

***Abstract*—The identification and classification of pests in rice field are the prerequisites of early warning systems for pest disasters. Among these pests, the rice planthoppers cause the most serious damage. However, the existing deep learning models for rice planthopper recognition are characterized by large size and numerous parameters, which makes them unsuitable for the deployment on embedded devices with limited computational resources. To address this issue, a lightweight rice planthopper recognition model based on YOLOv5 is proposed in this paper. In the model, a lightweight convolutional network named GhostNet is employed as the backbone to reduce the operational parameters. Additionally, a convolutional attention module (CBAM) is integrated into the backbone network to effectively enhance the transmission of deep information, so as to improve the model's ability of recognizing rice planthopper images. The original CIoU loss function is replaced by the SIoU loss function to expedite model convergence. Experimental results demonstrate that the modified model achieves the mAP@0.5 as 82.8%, with the parameter count of 3.12×10^6 and the model size of 7.2MB. Compared to the original model, it is a reduction of 46.7% in size and 43.3% in parameters, with a minor accuracy loss of 0.1%. Clearly, the improved model can achieve lightweight characteristics and robust performance, and thus it provides a theoretical and practical foundation for early warning systems against rice planthopper infestations in rice fields.**

***Index Terms*—rice planthopper, identification, YOLOv5, lightweight.**

I. Introduction

RICE, as one of the most crucial global food crops, holds a significant position in worldwide food production. However, rice faces threats from various pests during its growth. In there pests, the rice planthopper is a particularly significant menace due to its widespread distribution and formidable reproductive capabilities [1]. Traditional methods of monitoring rice planthoppers heavily rely on manual inspection, which causes challenges of low efficiency, insufficient coverage, and the tendency to overlook localized pest damage. As a result, these existing methods cannot meet the demands of modern agriculture for efficient and precise monitoring [2]. Thus, accurate identification of rice planthoppers is of paramount importance for precise prediction of rice pest situations, timely implementation of pest control measures, and the assurance of secure rice production.

In recent years, with the advancement and maturity of image recognition and deep learning algorithms, the pest detection methods based on image recognition and deep learning have wide applications in such field [3]. Ref. [4] proposed an improved method for rice pest image recognition based on the ResNet 34 model, which achieved an F_1 score of 0.98. Ref. [5] introduced a method by utilizing a stacked CNN architecture to significantly reduce model size, with an accuracy of 95% in rice pest recognition. Ref. [6] developed a new rice disease and pest recognition model based on the improved YOLOv7 algorithm, with the accuracy rates of 92.3% and mAP@0.5 of 93.7%.

Regarding the detection of rice planthoppers, Ref. [7] achieved a correct recognition rate of 91.7% based on four invariant moments and BP neural network. Ref. [8] proposed a dual-layer detection algorithm based on Faster R-CNN, with an average recall rate of 87.67% and an average accuracy of 97.36%. Ref. [9] suggested a support vector machine classification method by training on color and grayscale co-occurrence matrix image features, and obtained an accuracy of 87% in the classification of rice planthopper growth stages. It should be mentioned that these algorithms suffer the flaw of computationally intensive which results in bulky models and hinders their deployment on mobile devices. Therefore, on the foundation of ensuring accuracy in rice pest recognition, it is of significant importance to research into lightweight improvements for the existing deep learning algorithms.

Motivated by the aforementioned studies, this paper proposes a lightweight rice planthopper recognition method based on YOLOv5. In the method, on the basis of YOLOv5, we incorporates GhostNet into the backbone network, integrates the CBAM attention mechanism module, and replaces the YOLOv5 model's CIoU activation function with SIoU. This method ensures recognition accuracy while reducing model parameters and computational load, thereby it enhances the model's robustness and generalization capabilities.

Manuscript received February 5, 2024; revised July 4, 2024. This work was the National Natural Science Foundation of Jiangsu Province under grants BK20201469 and BE2021016-5, Postgraduate Research & Practice Innovation Program of Jiangsu Province under grants SJCX23_1181 and SJCX24_1288, and College Students Innovation and Entrepreneurship Training Program of Jiangsu Province under grant 202411276001Z.

Siquan Li is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail: y00450230122@njit.edu.cn).

Yi Wang is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (email: y00450220120@njit.edu.cn).

Teng Shi is a postgraduate student in the School of Mechanical Engineering, Nanjing Institute of Technology, Nanjing 211167, China (e-mail: y00450220231@njit.edu.cn).

Xi Chen is an undergraduate in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, P. R. China (e-mail: 455482577@qq.com).

Zhen Tang is an undergraduate in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, P. R. China (e-mail: 2142183890@qq.com).

Ziyu Zeng is an undergraduate in the School of Computer engineering, Nanjing Institute of Technology, Nanjing 211167, P. R. China (e-mail: 1695563372@qq.com).

Xin Wen is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail:y00450210233@njit.edu.cn).

Yanling Shang is an associate professor in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (corresponding author, e-mail:hnnhsyl@126.com).

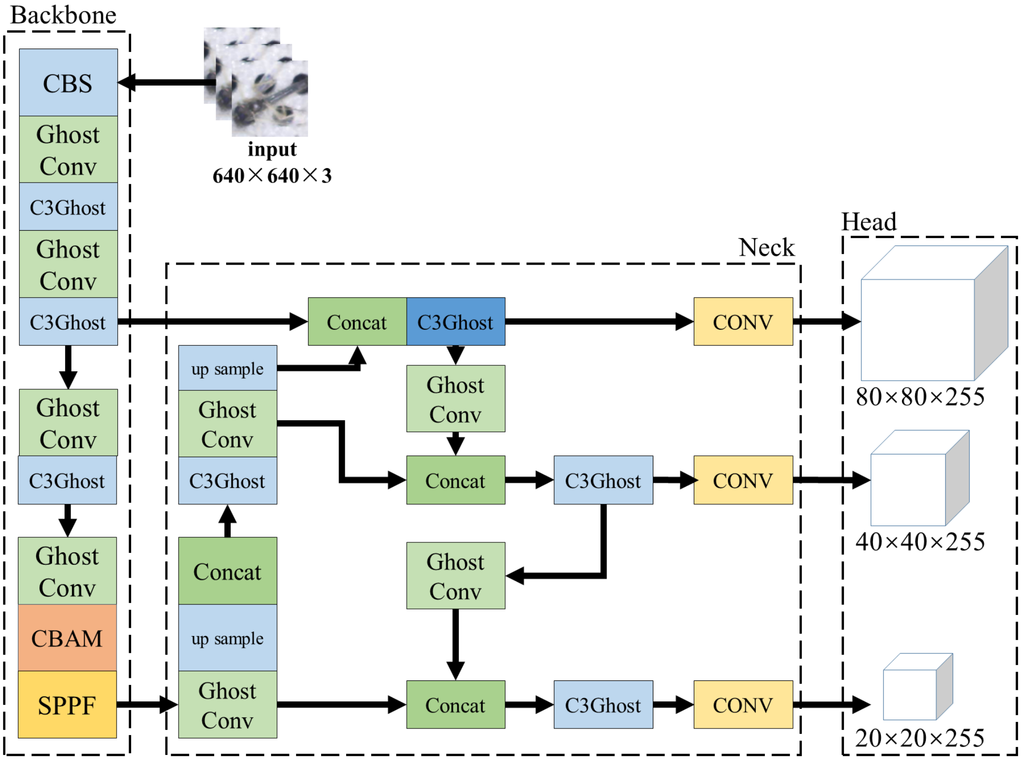

Fig. 1. Improved model architecture.

II. Network Model and Improvements

A. Image acquisition and dataset creation

Due to the lack of an available rice planthoppers dataset, this study opted to capture images of rice planthoppers using a pest lure platform at the Nanjing Taihe Rice Planting Cooperative. To minimize the interference of repetitive and substandard images with model training, the captured images underwent a screening process post-capture.

In this study, the open-source software LabelImg was employed for the annotation of rice planthopper images by utilizing a bounding box annotation method, and the labeled tags were designated as ”Rich PH.” Following the completion of annotations, the annotated dataset was divided into the training, testing, and validation sets in an ratio of 8:1:1.

B. Construction of rice planthoppers detection model

YOLO (You Only Look Once) is an object detection algorithm known for achieving real-time object detection [10]. Currently, YOLO has multiple versions, and YOLOv5 is the fifth version released in May 2020 which consists of four main components: input, backbone, neck, and head [11], [12]. The input handles image input, the Backbone extracts features from the input image, the neck integrates multi-scale feature maps more effectively, and the head outputs the position and category information of the target.

YOLOv5 is divided into five versions based on model size and complexity. YOLOv5m (Medium) [13], YOLOv5l (Large) [14], and YOLOv5x (Extra Large) [15] are relatively large models, unsuitable for embedded devices and scenarios demanding high real-time performance. YOLOv5n (Nano) [16] is too small, suitable only for some edge computing devices, with precision difficult to achieve at a significant level. Therefore, this paper selects YOLOv5s (Small) [17], which has relatively fewer parameters and faster speed, as the base algorithm for improvement.

As shown in Fig. 1, the overall framework of the improved model utilizes GhostNet to lightweight YOLOv5s. It combines the structurally complex and parameter-heavy C3 module with the Ghost Bottleneck module, forming the C3Ghost module. This module replaces all Bottleneck modules in the original model. Additionally, the Conv module in the original model is replaced with the Ghost module. To enhance the feature extraction capability of the model for small target sizes, the CBAM attention mechanism is integrated into the model, specifically in the last layer of the Backbone.

C. GhostNet architecture

The GhostNet architecture was introduced by the research team of the MIT Computer Science and Artificial Intelligence Laboratory in 2020 [18]. This architecture is a lightweight neural network designed specifically for mobile and edge devices with constrained computational resources. Its design philosophy aims to enhance efficiency by reducing model parameters and computational costs while maintaining high performance.

Compared to traditional structures, the advantage of the GhostNet network lies in the two-step generation of feature maps. Initially, GhostNet employs conventional convolutional operations to generate a portion of the feature map from input data. Subsequently, based on the generated feature map, GhostNet utilizes cheap operations (inexpensive linear transformations) to produce another portion of the feature map known as the Ghost feature map. Finally, GhostNet combines these two generated feature maps through concatenation and forms the ultimately required feature map. The

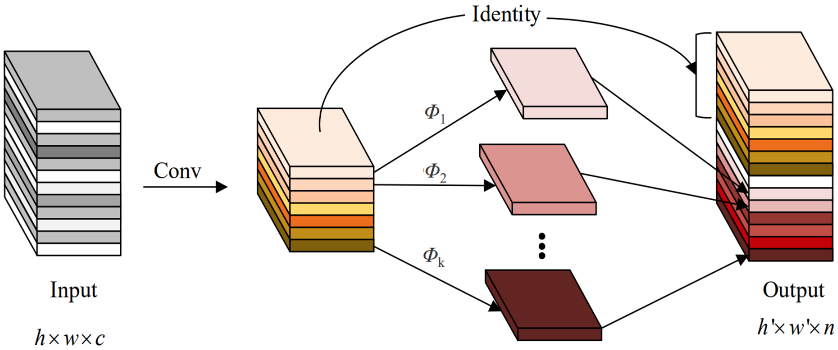

Fig. 2. GhostNet network structure.

structure diagram of the GhostNet network is illustrated in Fig. 2.

D. C3Ghost module

The Ghost Bottleneck module is a crucial structure within the GhostNet network. As illustrated in Fig. 3, it is comprising two distinct Ghost convolutions.

In the left diagram, the main section is composed of two Ghost modules (GM) connected in series. The first GM expands the channel count, and the second GM reduces the channel count to match the input channel count. Its function is to increase the depth of the network. In the right diagram, a deepwise convolution with a stride of 2 is introduced between the two GMs in the main section. This convolution compresses the height and width of the feature map, reducing it to half the size of the input. Its role is to decrease the matching channel count and connect the input and output of the two modules.

The C3Ghost module [19] follows the construction approach of the CSP module, consisting of two CBS modules and one Ghost Bottleneck module, as shown in Fig. 4. With this combination, the feature information of input is transformed into new feature maps through the CBS modules and Ghost Bottleneck module. This method not only reduces the overall model parameters but also maximally preserves the feature information.

E. CBAM Attention Mechanism

For the extraction of features from small targets, models often face interference from a significant amount of redundant background information. CBAM is a lightweight convolutional attention module that combines channel and spatial attention mechanisms, directing the model's focus more on the target object [20]. The structure of the CBAM attention module is illustrated in Fig. 5. It can be see that CBAM comprises two sub-modules: the channel attention mechanism (CAM) and the spatial attention mechanism (SAM) [21]. The channel attention mechanism is employed to focus on meaningful information in the rice planthopper images, while the spatial attention mechanism enables the model to concentrate attention on regions of interest. The structural diagrams of the CAM and SAM are presented in the Fig. 6 and Fig. 7.

CBAM sequentially applies channel attention and spatial attention to the input feature map, thereby it achieves attention adjustment in both the channel and spatial dimensions. This effectively emphasizes feature channels with high discriminative power for rice brown planthoppers, suppresses redundant and irrelevant features, and accurately extracts key features such as color, shape, and texture in rice planthopper images.

F. Improvement of loss function

The loss function is a tool for measuring the difference or error between the predicted values and the ground truth during the model training process [22]. Its objective is to minimize this difference, enabling the model to learn better and enhance performance [23]. Therefore, selecting an appropriate loss function plays a crucial role in the model's effectiveness.

The loss function employed by YOLOv5 is as follows:

$$loss = loss_{cls} + loss_{reg} + loss_{obj} \qquad (1)$$

In the equation, $loss$ represents the total loss, $loss_{cls}$,$loss_{reg}$ and $loss_{obj}$ represent the losses for category prediction, localization loss, and object presence probability loss. Both category prediction loss and object presence probability loss are calculated using binary cross-entropy loss, while the localization loss is computed using the CIoU loss function.

However, the CIoU loss function does not consider the mismatched orientation between the required and predicted bounding boxes, leading to slow convergence and inefficiency. Therefore, to address these issues, this paper adopts the SIoU loss function for calculating the localization loss. This loss function takes into account the angle of the vector between the required and predicted regressions, redefining the penalty metric, as illustrated in Fig. 8.

The formula for the SIoU loss function is as follows:

$$loss_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \qquad (2)$$

In the formula, Δ represents the distance cost, and Ω represents the shape cost. This loss function consists of three components: angle loss, distance loss, and shape loss. By considering the angle factor, during the regression process of the predicted box, it allows the predicted box to quickly regress to the same horizontal or vertical line as the ground

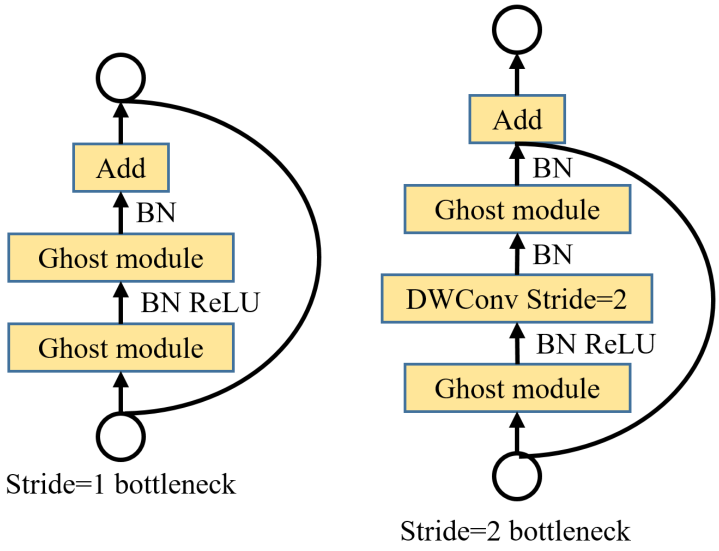

Fig. 3. Ghost Bottleneck network structure.

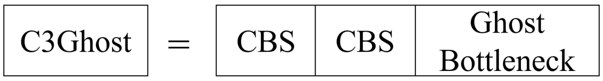

Fig. 4. C3Ghost network structure.

truth box, thereby accelerating the convergence speed of the loss function.

The specific calculation methods for the angle loss, distance loss, and shape loss in the SIoU loss function are as follows:

1) Angle loss: During the convergence process, it is first determined whether the angle is less than 45°. If it satisfies the condition, it is directly substituted into the following formula for calculation. Otherwise, its complementary angle is used instead. The formula for angle cost calculation is as follows:

$$\Lambda = 1 - 2\sin^2(\arcsin x - \frac{\pi}{4}) \tag{3}$$

$$x = \frac{c_h}{\sigma} = sin(\alpha) \tag{4}$$

In the formula, Λ represents the angle loss, x represents the sine value of α, σ represents the distance between the center points of the true box and the predicted box, and c_h represents the difference in the y-coordinate of the center points.

2) Distance loss: The calculation formula for the distance loss is as follows:

$$\Delta = \sum_{t=x,y} (1 - e^{\gamma\rho_t}) \tag{5}$$

In the formula, Δ represents the distance loss, $\gamma = 2 - \Lambda$ and ρ_t represents the squared difference between the center point coordinates of the two boxes.

3) Shape loss: The formula for shape loss is as follows:

$$\Omega = \sum_{t=w,h} (1 - e^{-\varpi_t})^\theta \tag{6}$$

In the formula, Ω represents the shape loss, θ represents the weight of the shape loss in the localization loss, and in this study, the parameter θ is set to the default value of 1. ϖ_w represents the ratio between the difference in width between the ground truth box and the predicted box and the maximum value, while ϖ_h represents the ratio between the difference in height between the ground truth box and the predicted box and the maximum value.

III. Results and Analysis

A. Experimental platform setup and hyperparameter configuration

The training platform utilized in this study is a computer equipped with the Windows 11 operating system. The CPU is an AMD Ryzen 7 6800H with Radeon Graphics, operating at a frequency of 3.20 GHz, with 64GB RAM. The GPU is an NVIDIA GeForce RTX3070Ti Laptop GPU. The training and testing environments are identical.

For the configuration of training parameters, the input image size is set to 640×640×3, the number of epochs is set to 200, the base learning rate is set to the default value of 1, the batch size is set to 8, the optimizer type is set to Stochastic Gradient Descent (SGD), the weight decay parameter is set to 5×10^{-4}. The deep learning framework used is PyTorch 1.10.0, and the Python version is 3.8.5.

B. Data enhancement

To enhance the model's generalization performance, Mosaic augmentation [24] and Mixup data augmentation are employed during the training. The Mosaic augmentation involves concatenating four small images that have undergone random cropping, scaling, and arrangement into one large image. This large image is then used as input for training, expanding the targets of small samples, enhancing the network's robustness, and improving its generalization ability. On the other hand, Mixup augmentation creates new training samples by linearly interpolating (mixing) the features and labels of two or more different samples during training. This process involves a weighted average of the pixel values and labels of two images, generating a new image and label. This method helps the model better handle boundaries between categories during training, thereby improving generalization

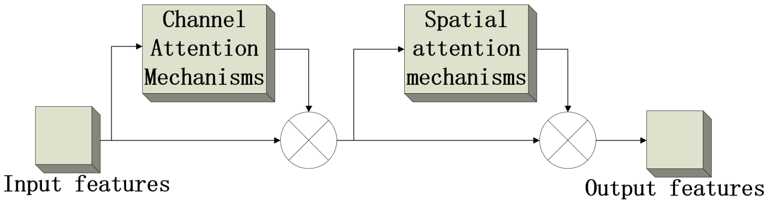

Fig. 5. CBAM attention module structure.

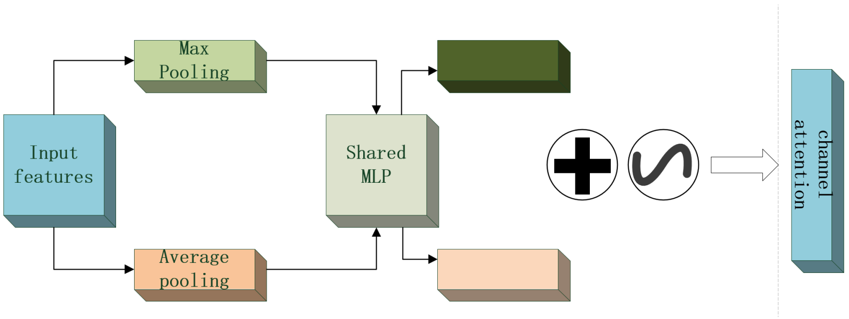

Fig. 6. Channel attention module structure.

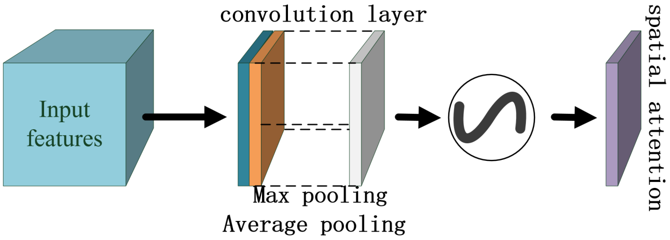

Fig. 7. Spatial attention module structure.

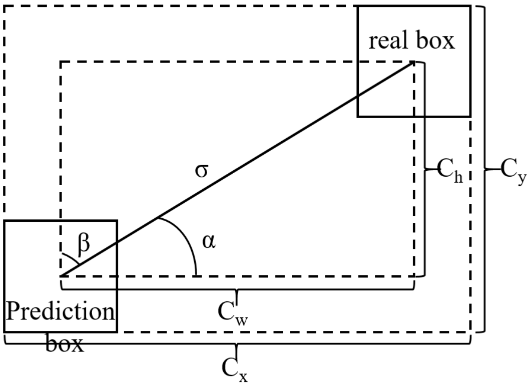

Fig. 8. Calculation of SIoU loss function

performance. Mixup also has the effect of reducing the model's overfitting to training samples.

The Mixup calculation formula is:

$$\widetilde{x} = \lambda x_i + (1 - \lambda) x_j \tag{7}$$

$$\widetilde{y} = \lambda y_i + (1 - \lambda) y_j \tag{8}$$

In the formula, (x_i, y_i) and (x_j, y_j) are random sample pairs from the original data, and λ is a parameter following a beta distribution with $\lambda \in [0, 1]$.

To assess the impact of data augmentation on the YOLOv5s base model used in this experiment, a comparison was made between the results with and without data augmentation. The comparative results are presented in Table I.

TABLE I
COMPARATIVE RESULTS BEFORE AND AFTER DATA AUGMENTATION

Model	mAP/%	F_1	Model Size/MB
Original Model	82.4	0.8	15.4
Data Augmented	82.9	0.8	15.4

From this table, it can be observed that the mAP value of the YOLOv5 model has improved with the use of data augmentation, which indicates that data augmentation can effectively enhance the model's average precision.

C. Evaluation metrics

Following the completion of training, the performance of the model needs to be evaluated. The performance evaluation metrics selected in this study primarily include Precision, Recall, F_1 Score, and mean Average Precision (mAP).

Prediction boxes with confidence scores greater than a threshold are defined as positive samples, otherwise, they are considered negative samples. The formulas for calculating Precision (P) and Recall (R) are as follows:

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

Here, P represents precision, R represents recall, TP is the number of examples the classifier correctly identifies as positive samples, FP is the number of examples the classifier incorrectly identifies as positive samples, and FN is the number of examples the classifier incorrectly identifies as negative samples.

F_1 is a metric for classification, calculated as the average of precision and recall. The formula is as follows:

$$F_1 = \frac{P+R}{2} \tag{11}$$

mAP@0.5 refers to the value of mAP when the threshold (score_threshold) is set to 0.5. It is calculated through the computation of precision (P) and pecall (R). The formula for calculating mAP is as follows:

$$mAP = \frac{1}{C}\sum_{k=i}^{n} P(k)\Delta R(k) \tag{12}$$

After training, the curves for Precision (P), Recall (R), F_1 Score (F_1), and mAP of the improved YOLOv5s model are shown in Figure 9.

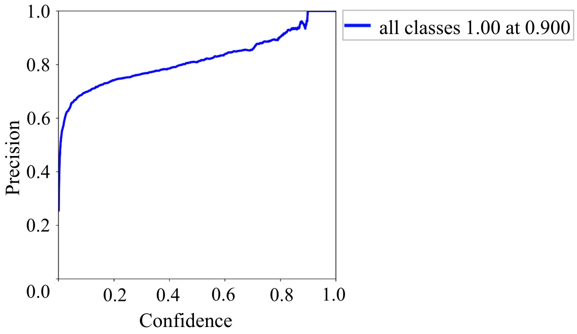

Fig. 9(a). Precision curve.

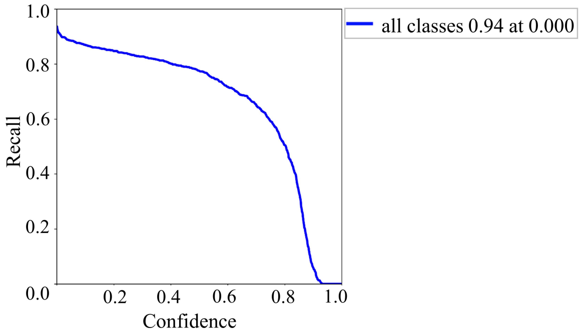

Fig. 9(b). Recall curve.

From the figures, it can be observed that with a threshold set at 0.5, the mAP@0.5 is 82.8%.

D. Ablation experiment results and analysis

To validate the performance of the improved YOLOv5s model, ablation experiments were conducted on the rice brown planthopper dataset. The experiments involved incorporating lightweight GhostNet modifications, CBAM attention mechanism enhancements, and SIoU loss function improvements onto the original YOLOv5s model. The results are presented in Table II.

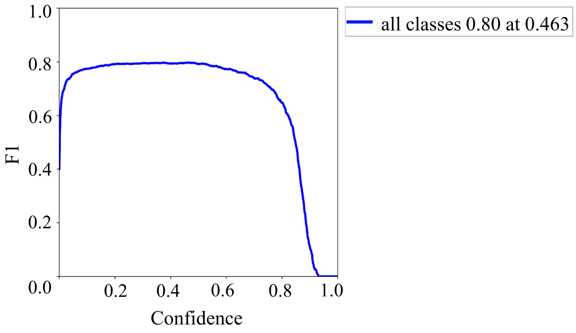

Fig. 9(c). F_1 score curve.

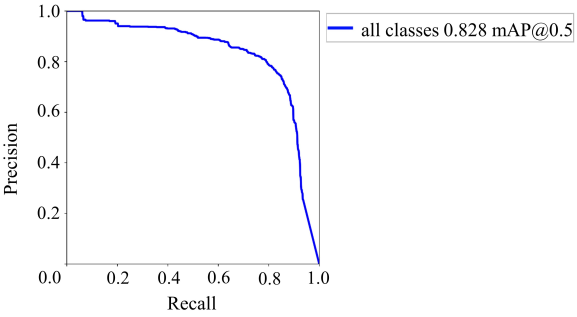

Fig. 9(d). mAP curve.

The experimental results demonstrate that the original YOLOv5s model achieves an mAP of 82.9% with a model size of 15.4MB. After lightweighting the model with the Ghost module, the model size is reduced to 7.1MB, representing a 46.7% reduction compared to the original size. Although the mAP value decreases by 1.5%, this reduction can be compensated for by adding the CBAM module and modifying the SIoU loss function. This indicates that the improved YOLOv5s model has a clear advantage and is effective for rice brown planthopper image recognition tasks. It can assist agricultural producers in rapidly and accurately detecting the quantity and distribution of rice brown planthoppers, enabling the implementation of effective control measures.

E. Cross-comparison experiment

In order to objectively evaluate the performance of our model, this study trained other models under the same conditions and compared them with our model. The results are shown in Table III. It can be observed that compared to the popular lightweight network models YOLOv3-tiny and YOLOv4-tiny, our model achieves an improvement in mAP by 8.2% and 5.5%, respectively, with a significant reduction in parameters and model size. Compared to the mainstream detection model SSD, our model shows a substantial increase in accuracy and a considerable reduction in parameters and model size. In comparison to the original YOLOv5s model, although there is a slight 0.1% mAP loss, the parameters and model size are significantly reduced by 46.7% and 43.3%, respectively.

IV. Conclusion

To deal with the limitations of traditional deep learning classification methods in rice planthopper recognition, this

TABLE II
COMPARATIVE RESULTS BEFORE AND AFTER DATA AUGMENTATION

Ghost Module	CBAM Module	SIoU loss	mAP/%	Model Size/MB
			82.9	15.4
✓			81.4	7.1
✓	✓		82.5	7.2
✓	✓	✓	82.8	7.2

TABLE III
COMPARATIVE RESULTS OF DIFFERENT MODELS

Model	mAP/%	Parameters/10^6	Model Size/MB
SSD	61.2	24.5	87.4
YOLOv3-tiny	74.6	9.11	17.0
YOLOv4-tiny	77.3	5.91	21.7
YOLOv5s	82.9	7.2	15.4
improved algorithm	82.8	3.12	7.2

study has proposed a lightweight rice planthopper recognition model based on YOLOv5. The model has developed by integration of the GhostNet network structure, CBAM attention mechanism, and replacement of the SIoU loss function in the YOLOv5s network. The experiment results show that the proposed model with pest images yields an mAP@0.5 value of 82.8%. Compared to the original model, the accuracy loss is only 0.1%, while the parameters and model size are significantly streamlined, which makes it more suitable for deployment on embedded devices. As a result, the achievements of this study not only hold positive implications for the sustainable development of rice production but also bear significant value in advancing the level of agricultural intelligence. And thus it reduces agricultural production costs, and enhances agricultural productivity.

REFERENCES

[1] O. Azzam and T. C. Chancellor, "The biology, epidemiology, and management of rice tungro disease in asia," *Plant Disease*, vol. 86, no. 2, pp. 88–100, 2002.

[2] N. Krishnaiah, "A global perspective of rice brown planthopper management iii-strategies for bph management," *Rice Genomics and Genetics*, vol. 5, pp. 1–11, 2014.

[3] S. Verma, S. P. Sahu, and T. P. Sahu, "Stock market forecasting with different input indicators using machine learning and deep learning techniques: A review," *Engineering Letters*, vol. 31, no. 1, pp. 213–229, 2023.

[4] J. Wang, X. Liang, and G. Lei, "Image recognition of rice diseases and insect pests based on deep residual network and transfer learning," *Journal of Chinese Agricultural Mechanization*, vol. 179, p. 105834, 2020.

[5] C. R. Rahman, P. S. Arko, M. E. Ali, M. A. I. Khan, S. H. Apon, F. Nowrin, and A. Wasif, "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosystems Engineering*, vol. 194, pp. 112–120, 2020.

[6] L. Jia, T. Wang, Y. Chen, Y. Zang, X. Li, H. Shi, and L. Gao, "Mobilenet-ca-yolo: An improved yolov7 based on the mobilenetv3 and attention mechanism for rice pests and diseases detection," *Agriculture*, vol. 13, no. 7, p. 1285, 2023.

[7] X. Zou, W. Ding, D. Liu, and S. Zhao, "Classification of rice planthopper based on invariant moments and bp neural network," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 29, no. 18, pp. 171–178, 2013.

[8] Y. He, Z. Zhou, L. Tian, Y. Liu, and X. Luo, "Brown rice planthopper (nilaparvata lugens stal) detection based on deep learning," *Precision Agriculture*, vol. 21, no. 6, pp. 1385–1402, 2020.

[9] S. Watcharabutsarakham and I. Methasate, "Mobile-device based image processing for rice brown planthopper classification and outbreak monitoring," *Applied Engineering in Agriculture*, vol. 35, no. 1, pp. 15–21, 2019.

[10] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[11] W. Zhang, Y. Zhao, Y. Guan, T. Zhang, Q. Liu, and W. Jia, "Green apple detection method based on optimized yolov5 under orchard environment," *Engineering Letters*, vol. 31, no. 3, pp. 1104–1113, 2023.

[12] Y. Liu and Y. Tian, "Dcms-yolov5: A dual-channel and multi-scale vertical expansion helmet detection model based on yolov5," *Engineering Letters*, vol. 31, no. 1, pp. 373–379, 2023.

[13] S. Luo and J. Liu, "Research on car license plate recognition based on improved yolov5m and lprnet," *IEEE Access*, vol. 10, pp. 93 692–93 700, 2022.

[14] G. Li, L. Fu, C. Gao, W. Fang, G. Zhao, F. Shi, J. Dhupia, K. Zhao, R. Li, and Y. Cui, "Multi-class detection of kiwifruit flower and its distribution identification in orchard based on yolov5l and euclidean distance," *Computers and Electronics in Agriculture*, vol. 201, p. 107342, 2022.

[15] J. Zhang, W. Su, H. Zhang, and Y. Peng, "Se-yolov5x: An optimized model based on transfer learning and visual attention mechanism for identifying and localizing weeds and vegetables," *Agronomy*, vol. 12, no. 9, p. 2061, 2022.

[16] Y. Liu, Y. Wang, Y. Li, Q. Li, and J. Wang, "Setr-yolov5n: A lightweight low-light lane curvature detection method based on fractional-order fusion model," *IEEE Access*, vol. 10, pp. 93 003–93 016, 2022.

[17] X. Zhang and Y. Tia, "Improved yolov5s traffic sign detection," *Engineering Letters*, vol. 31, no. 4, pp. 1883–1893, 2023.

[18] M. E. Paoletti, J. M. Haut, N. S. Pereira, J. Plaza, and A. Plaza, "Ghostnet for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 12, pp. 10 378–10 393, 2021.

[19] J. Xu, H. Yang, Z. Wan, H. Mu, D. Qi, and S. Han, "Wood surface defects detection based on the improved yolov5-c3ghost with simam module," *IEEE Access*, vol. 11, no. 12, pp. 105 281–105 287, 2023.

[20] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "Cbam: convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[21] Y. Wang, Y. Duan, and H. Wu, "Image defogging algorithm based on attention mechanism and split convolution," *Engineering Letters*, vol. 31, no. 4, pp. 1567–1573, 2023.

[22] B. Zhang, X. Zhang, and Z. Li, "An efficient face mask wearing detection algorithm based on improved yolov3," *Engineering Letters*, vol. 30, no. 4, pp. 1493–1503, 2022.

[23] X. Wen, Y. Yao, Y. Cai, Z. Zhao, T. Chen, Z. Zeng, Z. Tang, and F. Gao, "A lightweight st-yolo based model for detection of tea bud in unstructured natural environments," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 3, pp. 342–349, 2024.

[24] H. Wang and Z. Song, "Improved mosaic: Algorithms for more complex images," in *Journal of Physics: Conference Series*, vol. 1684, no. 1. IOP Publishing, 2020, p. 012094.